REMARKS

This is in response to the Office Action of May 2, 2003. Claims 13-24 were cancelled without prejudice. Claims 25- 44 were added. Claims 25-44 are pending.

New independent claims 25 and 30 recite limitations for a graphics system having a polygon rendering graphics module that includes a cache and a cache controller. Support for claims 25 and 29 is found throughout Applicant's specification, such as in FIG. 1, FIG. 9, page 5, lines 20-23, and page 14. New independent method claims 37 and 41 recite method limitations for a method of using a cache in a graphics module to reduce data transfers across a system bus and are similarly supported.

The Examiner rejected claims 13-17 under 35 USC § 112 as being indefinite. In light of the cancellation of claims 13-17 and the submission of new claims, it is believed that this rejection has been addressed, since the new claims do not have the configuration of elements that the examiner contended was indefinite.

The Examiner rejected claims 13-16, 18, 19, 21 and 23 under 35 U.S.C. 103(a) as being unpatentable over Morgan, et al. (U.S. Pat. No. 5,821,940). Claims 17, 22, and 24 were rejected under 35 U.S.C. 103(a) over Morgan in view of Holt (U.S. Pat. No. 5,760,792). Claim 20 was rejected under 35 U.S.C. over Morgan in view of Swanson (U.S. Pat. No. 5,421,028).

One benefit of Applicant's claimed invention is that it reduces the number of data transfers required to render images. A data transfer of vertex data across a system bus can require a substantial number of bytes of information, which may necessitate a substantial bus bandwidth and/or require multiple data transfers. (See, e.g., Applicant's specification on page 2, line 23 to page 3, line 12). In Applicant's claimed invention, vertex data is cached in the graphics module. As a result, only vertex data for vertices that are not already cached needs a memory transfer across the system bus, providing a reduction in bandwidth and/or the number of data transfers required to fulfill commands from the CPU to render polygons. Moreover, the cache may be updated after memory transfers. This permits, for example, updates stored in the cache for one vertex to be reused for other polygons having the vertex in common.

Applicant respectfully submits that the cited art does not teach or suggest Applicant's claimed invention. Morgan is directed to the problem of reducing the computation time required to perform matrix calculations for rotating displayed images in a flight training simulator

(Morgan, column 1, lines 7-11, lines 21-22, and lines 59-66). The matrix calculations are performed for each vertex by a transformation processor 18. Matrix calculations performed for a vertex are cached in cache 22. Caching the output of transformation processor 18 eliminates the need to perform a matrix calculation more than once for vertices shared in common by two or more polygons (Morgan, column 2, lines 2-5 and lines 32-39). The matrix-transformed vertex data is sent over a system bus to a backend processor 24 that performs scan line conversion and pixel rendering for the rotated images.

However, the cache in Morgan is located on the CPU side of the system bus in order to receive the output of the transformation processor. Consequently, Morgan sends all transformed vertex data, whether cached or uncached, to the backend processor 24 on a single system bus 31. (See, e.g., Morgan at column 4, lines 36-42, column 5, lines 17-20, and FIGS. 5-6). As a result, the cache in Morgan does not reduce the number of data transfers of vertex data that must be sent across the system bus to the backend processor.

Applicant respectfully submits that Morgan does not teach or suggest the claim limitations of claims 25, 30, 37, and 41 corresponding to a cache for vertex data that is local to a graphics module, thereby reducing transfers of vertex data across a system bus required to render polygons. There is no teaching or suggestion of any kind in Morgan for placing a cache within a graphics module. In contrast, the cache in Morgan is located on the CPU side of the system bus, necessitating that all vertex data, whether cached or uncached, be sent through the system bus to the backend processor.

Morgan also does not teach or suggest a cache controller residing within a graphics module as required by claims 25 and 30 for checking a local cache. Similarly, Morgan does not teach or suggest checking a local cache, as required by claims 37 and 41. In contrast, Morgan has its cache control elements located on the CPU side of the system bus.

In regards to claim 30, which recites a direct memory access (DMA) engine limitation, Applicant notes that Holt does not teach or suggest a cache controller using a DMA module to selectively transfer vertex data not present in the cache. In contrast, Holt teaches using DMA to transfer all vertex data from main memory to a graphics FIFO. (Holt, column 2, lines 56-61).

The dependent claims are allowable for at least the same reasons as those described above in regards to the independent claims. Additionally, the dependent claims recite additional limitations not taught or suggested by the cited references.

In view of the foregoing amendments and remarks, it is respectfully submitted that the application is now in a condition for allowance. The Examiner is invited to contact the undersigned if there are any residual issues that can be resolved through a telephone call.

The Commissioner is hereby authorized to charge any appropriate fees to Deposit Account No 03-3117.

Dated: 7/18/03

Respectfully submitted,
COOLEY GODWARD LLP

By: Edward Van Gieson

Edward A. Van Gieson
Reg. No. 44,386

Cooley Godward LLP
ATTN: Patent Group
Five Palo Alto Square
3000 El Camino Real
Palo Alto, CA 94306-2155
Tel: (650) 843-5000
Fax: (650) 857-0663